REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
	Technical Report	-

4. TITLE AND SUBTITLE
A Priori Error-Controlled Simulation of Electromagnetic Phenomena for HPC

5a. CONTRACT NUMBER

5b. GRANT NUMBER
W911NF-14-C-0011

5c. PROGRAM ELEMENT NUMBER
665502

6. AUTHORS
Xing He, Ramakanth Munipalli, Thomas Hagstrom

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
HyPerComp, Inc.
2629 Townsgate Road
Suite 105
Westlake Village, CA 91361 -2981

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
64503-MA-ST2.1

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
In this project we aim to construct a high fidelity boundary condition module for Maxwell's equations that can be interfaced with major time-domain electromagnetics solver systems. There is ample need in the EM modeling community for reliable and stable far field boundary conditions of high accuracy. Most existing methods are limited in one or more of these requirements, and recent developments in the CRBC procedure (as originally presented by Hagstrom and Warburton in 2009), have made the technique an attractive candidate for implementation in multi-purpose solvers. In phase-I of this project

15. SUBJECT TERMS
Computational electromagnetics, time-domain, outer boundary conditions, high order accuracy, modular software

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ramakanth Munipalli
UU	UU	UU	UU		19b. TELEPHONE NUMBER 805-371-7500

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

A Priori Error-Controlled Simulation of Electromagnetic Phenomena for HPC

ABSTRACT

In this project we aim to construct a high fidelity boundary condition module for Maxwell's equations that can be interfaced with major time-domain electromagnetics solver systems. There is ample need in the EM modeling community for reliable and stable far field boundary conditions of high accuracy. Most existing methods are limited in one or more of these requirements, and recent developments in the CRBC procedure (as originally presented by Hagstrom and Warburton in 2009), have made the technique an attractive candidate for implementation in multi-purpose solvers. In phase-I of this project we implemented and improved upon many aspects of this method, particularly in light of the needs of high order accurate Maxwell equations solvers (based on the discontinuous Galerkin method). Error bounds were computed and demonstrated for a number of cases. We continue in the second phase of this project to improve upon the robustness of this method, as we develop a software platform which shall be its flagship (and open source) implementation. In this first quarterly report we present initial progress to this end, showing recent mathematical developments, project coordination plans and some results from our initial experience with MEEP.

Cover Page

Contract Number:	**W911NF-13-C-0039**
Proposal Number:	**A2-5030**
Contractor's Name and Address:	HyPerComp, Inc. 2629 Townsgate Road, Suite 105 Westlake Village, CA 91361
Title of the Project:	**A Priori Error-Controlled Simulation of Electromagnetic Phenomena for HPC**
Contract Performance Period:	March 13, 2013 - March 12, 2015
Current Reporting Period:	March 13, 2013 - June 12, 2013

Total Contract Amount:	$749,996
Amount of funds paid by DFAS to date:	$0
Total amount expended/invoiced to date:	$0
Number of employees working on the project:	3
Number of new employees placed on contract this month:	0

Contents

1 Abstract

In this project we aim to construct a high fidelity boundary condition module for Maxwell's equations that can be interfaced with major time-domain electromagnetics solver systems. There is ample need in the EM modeling community for reliable and stable far field boundary conditions of high accuracy. Most existing methods are limited in one or more of these requirements, and recent developments in the CRBC procedure (as originally presented by Hagstrom and Warburton in 2009), have made the technique an attractive candidate for implementation in multi-purpose solvers. In phase-I of this project we implemented and improved upon many aspects of this method, particularly in light of the needs of high order accurate Maxwell equations solvers (based on the discontinuous Galerkin method). Error bounds were computed and demonstrated for a number of cases. We continue in the second phase of this project to improve upon the robustness of this method, as we develop a software platform which shall be its flagship (and open source) implementation. In this first quarterly report we present initial progress to this end, showing recent mathematical developments, project coordination plans and some results from our initial experience with MEEP.

2 Introduction

In this project, HyPerComp is collaborating with Prof. Thomas Hagstrom and his research group at the Southern Methodist University (SMU). Roles of the two organizations are very broadly divided into mathematical method development (led by SMU) and implementation, software development and maturation (led by HyPerComp). The project is coordinated via a series of in-person and telephone meetings. As of May 17, 2013 we have been conducting weekly telephone meetings. Two students, John Lagrone and Fritz Juhnke have been included in the team and have been actively participating in the work so far.

Tasks: The following is a list of tasks to be performed in this project.

1. Project Formulation
2. Software Development
3. Verification Validation
4. Coupling
5. Efficiency Testing
6. Release of software
7. Documentation
8. Sustainability Plan
9. User Support

At present, we are working on a refined formulation of the CRBC method and testing it in sample problems. Primary concerns pertaining to method stability at corners, particularly in 3D are being addressed. CRBC implementations in finite difference schemes, DG (in FORTRAN as well as in MATLAB) are available from prior research in this project, for testing. We hope to host a telephone meeting with our project manager from the Army in July, coinciding with Prof. Hagstrom's visit to HyPerComp.

We are presently aiming to integrate the CRBC module with the following codes:

- HDphysics from HyPerComp, a high order DG based solver
- MEEP from MIT, an open source FDTD code
- cgmx part of "Overture" suite of simulation codes from LLNL - high order finite differences, second order PDEs
- CLAWPACK a finite difference suite of solvers from U.Washington

Students from SMU shall initially focus on an FDTD implementation with MEEP and begin to identify software needs. We are in the process of developing software requirements for each of the systems mentioned above, so that we can outline a common implementation of the method and programming techniques. This shall be discussed in the forthcoming report.

3 CRBCs and Corner Conditions for the TM Maxwell System

Consider the TM Maxwell system:

$$\frac{\partial H^x}{\partial t} + \frac{1}{\mu}\frac{\partial E^z}{\partial y} = 0 \tag{1}$$

$$\frac{\partial H^y}{\partial t} - \frac{1}{\mu}\frac{\partial E^z}{\partial x} = 0 \tag{2}$$

$$\frac{\partial E^z}{\partial t} - \frac{1}{\epsilon}\frac{\partial H^y}{\partial x} + \frac{1}{\epsilon}\frac{\partial H^x}{\partial y} = 0 \tag{3}$$

and set $c = \frac{1}{\sqrt{\epsilon\mu}}$.

CRBC on an arbitrary edge

Consider a portion of the radiation boundary with unit normal n pointing **outward** from the computational domain and unit tangent vector τ:

$$n = \begin{pmatrix} n_x \\ n_y \end{pmatrix}, \quad \tau = \begin{pmatrix} -n_y \\ n_x \end{pmatrix}. \tag{4}$$

(Notice we have chosen a specific orientation for simplicity.) Introducing angles ϕ_j, $\bar{\phi}_j$, $j = 1, \ldots, P$, we rewrite the Maxwell system in terms of normal and tangential derivatives, $\frac{\partial}{\partial n}$ and $\frac{\partial}{\partial \tau}$ and recursively replace normal derivatives with interpolation operators. For outgoing waves

$$\frac{\partial}{\partial n} \approx -\frac{\cos\phi_j}{c}\frac{\partial}{\partial t} - \frac{\sin^2\phi_j}{cT\cos\phi_j}, \tag{5}$$

and for incoming waves

$$\frac{\partial}{\partial n} \approx \frac{\cos\bar{\phi}_j}{c}\frac{\partial}{\partial t} + \frac{\sin^2\bar{\phi}_j}{cT\cos\bar{\phi}_j}. \tag{6}$$

Although not necessary, it is convenient to recast the first two equations using the normal and tangential components of H:

$$\frac{\partial}{\partial t}\left(n_x H^x + n_y H^y\right) + \frac{1}{\mu}\frac{\partial E^z}{\partial \tau} = 0 \tag{7}$$

$$\frac{\partial}{\partial t}\left(-n_y H^x + n_x H^y\right) - \frac{1}{\mu}\frac{\partial E^z}{\partial n} = 0 \tag{8}$$

Introducing auxiliary variables H_j^x, H_j^y, and E_j^z with H_0^x, H_0^y, and E_0^z coinciding with the trace of the solution at the boundary (or in the DG context the boundary state) we solve for $j = 1, \ldots P$

$$\begin{aligned} \frac{\partial}{\partial t}\left(-n_y H_{j-1}^x + n_x H_{j-1}^y\right) + \frac{\cos\phi_j}{\mu c}\frac{\partial E_{j-1}^z}{\partial t} + \frac{1}{\mu cT}\frac{\sin^2\phi_j}{\cos\phi_j}E_{j-1}^z = \\ \frac{\partial}{\partial t}\left(-n_y H_j^x + n_x H_j^y\right) - \frac{\cos\bar{\phi}_j}{\mu c}\frac{\partial E_j^z}{\partial t} - \frac{1}{\mu cT}\frac{\sin^2\bar{\phi}_j}{\cos\bar{\phi}_j}E_j^z \end{aligned} \tag{9}$$

$$\begin{aligned}\frac{\partial E^z_{j-1}}{\partial t} + \frac{\cos\phi_j}{\epsilon c}\frac{\partial}{\partial t}\left(-n_y H^x_{j-1} + n_x H^y_{j-1}\right) \\ +\frac{1}{\epsilon c T}\frac{\sin^2\phi_j}{\cos\phi_j}\left(-n_y H^x_{j-1} + n_x H^y_{j-1}\right) + \frac{1}{\epsilon}\frac{\partial}{\partial \tau}\left(n_x H^x_{j-1} + n_y H^y_{j-1}\right) \quad = \\ \frac{\partial E^z_j}{\partial t} - \frac{\cos\bar{\phi}_j}{\epsilon c}\frac{\partial}{\partial t}\left(-n_y H^x_j + n_x H^y_j\right) \\ -\frac{1}{\epsilon c T}\frac{\sin^2\bar{\phi}_j}{\cos\bar{\phi}_j}\left(-n_y H^x_j + n_x H^y_j\right) + \frac{1}{\epsilon}\frac{\partial}{\partial \tau}\left(n_x H^x_j + n_y H^y_j\right),\end{aligned} \tag{10}$$

and for $j = 0, \ldots, P$

$$\frac{\partial}{\partial t}\left(n_x H^x_j + n_y H^y_j\right) + \frac{1}{\mu}\frac{\partial E^z_j}{\partial \tau} = 0. \tag{11}$$

These are $3P+1$ equations for $3P+3$ unknowns. Two additional equations are obtained first by using data from the interior for outgoing characteristics and terminating the incoming characteristic recursion. The system simplifies if we introduce the normal characteristic variables:

$$R_{\pm,j} = E^z_j \pm \sqrt{\frac{\mu}{\epsilon}}\left(-n_y H^x_j + n_x H^y_j\right), \quad H_{n,j} = n_x H^x_j + n_y H^y_j, \tag{12}$$

then the recursions take the form

$$\begin{aligned}(1+\cos\phi_j)\frac{\partial R_{+,j-1}}{\partial t} + \frac{1}{T}\frac{\sin^2\phi_j}{\cos\phi_j}R_{+,j-1} + \frac{1}{\epsilon}\frac{\partial H_{n,j-1}}{\partial \tau} \quad = \\ (1-\cos\bar{\phi}_j)\frac{\partial R_{+,j}}{\partial t} - \frac{1}{T}\frac{\sin^2\bar{\phi}_j}{\cos\bar{\phi}_j}R_{+,j} + \frac{1}{\epsilon}\frac{\partial H_{n,j}}{\partial \tau}.\end{aligned} \tag{13}$$

$$\begin{aligned}(1-\cos\phi_j)\frac{\partial R_{-,j-1}}{\partial t} - \frac{1}{T}\frac{\sin^2\phi_j}{\cos\phi_j}R_{-,j-1} + \frac{1}{\epsilon}\frac{\partial H_{n,j-1}}{\partial \tau} \quad = \\ (1+\cos\bar{\phi}_j)\frac{\partial R_{-,j}}{\partial t} + \frac{1}{T}\frac{\sin^2\bar{\phi}_j}{\cos\bar{\phi}_j}R_{-,j} + \frac{1}{\epsilon}\frac{\partial H_{n,j}}{\partial \tau}.\end{aligned} \tag{14}$$

We then take

$$\frac{\partial R_{-,0}}{\partial t} = \left(\frac{\partial R_{-,0}}{\partial t}\right)^{\text{interior}}, \tag{15}$$

and solve (14) in increasing j for $\frac{\partial R_{-,j}}{\partial t}$. The termination condition is

$$R_{+,P} = 0, \tag{16}$$

which allows us to solve (13) in decreasing j for $\frac{\partial R_{+,j-1}}{\partial t}$.

Corner

We now consider the case of two artificial boundaries meeting at a corner point. Let $n^1 = (n^1_x, n^1_y)^T$ be the unit outward normal for the first boundary and $n^2 = (n^2_x, n^2_y)^T$ be the unit outward normal for the second. Introduce

$$\gamma = n^1 \cdot n^2, \quad \gamma^2 < 1.$$

Notice that if the angle between the edges is obtuse then $\gamma > 0$. Also introduce $s_j = \pm 1$, $j = 1, 2$ with $s_j = 1$ if the edge orientation is **into** the corner and $s_j = -1$ otherwise. We now solve for a doubly-indexed array of auxiliary variables $E^z_{j,k}$, $H^x_{j,k}$ and $H^y_{j,k}$ which coincide with the corner values of the auxiliary variables $E^{1,z}_k$, $H^{1,x}_k$, $H^{1,y}_k$ when $j = 0$ and $E^{2,z}_j$, $H^{2,x}_j$, $H^{2,y}_j$ when $k = 0$. Thus the auxiliary variables with index $0, 0$ correspond to the corner values of the actual fields. A complete set of equations for the corner variables is obtained by applying the interpolation conditions to replace all space derivatives. We do so by solving for ∇W in terms of $n^1 \cdot \nabla W$ and $n^2 \cdot \nabla W$ where W is any function. This yields

$$\begin{aligned} \nabla W &= (1-\gamma^2)^{-1}(n^1 - \gamma n^2)(n^1 \cdot \nabla W) \\ &\quad +(1-\gamma^2)^{-1}(n^2 - \gamma n^1)(n^2 \cdot \nabla W) \end{aligned} \tag{17}$$

To further simplify the formula we use the fact that $\|n^1\| = \|n^2\| = 1$ and find that

$$\begin{aligned} 1-\gamma^2 &= 1 - (n^1_x n^2_x + n^1_y n^2_y)^2 \\ &= \left((n^1_x)^2 + (n^1_y)^2\right)\left((n^2_x)^2 + (n^2_y)^2\right) - (n^1_x n^2_x + n^1_y n^2_y)^2 \\ &= (n^1_x)^2 (n^2_y)^2 + (n^1_y)^2 (n^2_x)^2 - 2 n^1_x n^2_x n^1_y n^2_y \\ &= S^2 \end{aligned} \tag{18}$$

where we have introduced

$$S = n^1_x n^2_y - n^1_y n^2_x. \tag{19}$$

Then

$$\begin{aligned} (n^1 - \gamma n^2) &= \begin{pmatrix} n^1_x - n^1_x (n^2_x)^2 - n^1_y n^2_y n^2_x \\ n^1_y - n^1_x n^2_x n^2_y - n^1_y (n^2_y)^2 \end{pmatrix} \\ &= \begin{pmatrix} n^1_x (n^2_y)^2 - n^1_y n^2_y n^2_x \\ n^1_y (n^2_x)^2 - n^1_x n^2_x n^2_y \end{pmatrix} \\ &= S \begin{pmatrix} n^2_y \\ -n^2_x \end{pmatrix} \end{aligned} \tag{20}$$

Similarly

$$(1-\gamma^2)^{-1}(n^2 - \gamma n^1) = \frac{1}{S}\begin{pmatrix} -n^1_y \\ n^1_x \end{pmatrix}. \tag{21}$$

Thus

$$\frac{\partial}{\partial x} = \frac{1}{S}\left(n^2_y \frac{\partial}{\partial n^1} - n^1_y \frac{\partial}{\partial n^2}\right) \tag{22}$$

$$\frac{\partial}{\partial y} = \frac{1}{S}\left(-n^2_x \frac{\partial}{\partial n^1} + n^1_x \frac{\partial}{\partial n^2}\right). \tag{23}$$

Then all derivatives can be replaced by the interpolation conditions (5)-(6) associated with one of the edges. We also assume for simplicity that we are using the same angle parameters and orders on each edge. This is not necessary, but it is what we usually do and assuming it simplifies the notation.

The recursions corresponding to (9) are the easiest to write down. We solve for $j = 1, \ldots, P$, $k = 0, \ldots, P$

$$\begin{aligned} &\frac{\partial}{\partial t}\left(-n^2_y H^x_{j-1,k} + n^2_x H^y_{j-1,k}\right) + \frac{\cos\phi_j}{\mu c}\frac{\partial E^z_{j-1,k}}{\partial t} + \frac{1}{\mu c T}\frac{\sin^2\phi_j}{\cos\phi_j} E^z_{j-1,k} = \\ &\qquad \frac{\partial}{\partial t}\left(-n^2_y H^x_{j,k} + n^2_x H^y_{j,k}\right) - \frac{\cos\bar{\phi}_j}{\mu c}\frac{\partial E^z_{j,k}}{\partial t} - \frac{1}{\mu c T}\frac{\sin^2\bar{\phi}_j}{\cos\bar{\phi}_j} E^z_{j,k}, \end{aligned} \tag{24}$$

and for $j = 0, \ldots, P$, $k = 1, \ldots, P$

$$\frac{\partial}{\partial t}\left(-n_y^1 H_{j,k-1}^x + n_x^1 H_{j,k-1}^y\right) + \frac{\cos\phi_k}{\mu c}\frac{\partial E_{j,k-1}^z}{\partial t} + \frac{1}{\mu c T}\frac{\sin^2\phi_k}{\cos\phi_k}E_{j,k-1}^z = \frac{\partial}{\partial t}\left(-n_y^1 H_{j,k}^x + n_x^1 H_{j,k}^y\right) - \frac{\cos\bar{\phi}_k}{\mu c}\frac{\partial E_{j,k}^z}{\partial t} - \frac{1}{\mu c T}\frac{\sin^2\bar{\phi}_k}{\cos\bar{\phi}_k}E_{j,k}^z. \tag{25}$$

Lastly from the (10) we derive a relationship involving both recursions which we impose for $j, k = 1, \ldots, P$. Using (22)-(23) to express the space derivatives using the normals we find

$$\epsilon S\frac{\partial E^z}{\partial t} - n_y^2\frac{\partial H^y}{\partial n^1} + n_y^1\frac{\partial H^y}{\partial n^2} - n_x^2\frac{\partial H^x}{\partial n^1} + n_x^1\frac{\partial H^x}{\partial n^2} = 0. \tag{26}$$

Now replacing the normal derivatives by (5)-(6) we derive:

$$\begin{aligned}
&S\frac{\partial}{\partial t}\left(E_{j-1,k-1}^z + E_{j,k}^z - E_{j,k-1}^z - E_{j-1,k}^z\right) \\
&+\frac{\cos\phi_j}{\epsilon c}\frac{\partial}{\partial t}\left(-n_x^1\left(H_{j-1,k-1}^x - H_{j-1,k}^x\right) - n_y^1\left(H_{j-1,k-1}^y - H_{j-1,k}^y\right)\right) \\
&+\frac{\sin^2\phi_j}{\epsilon c T\cos\phi_j}\left(-n_x^1\left(H_{j-1,k-1}^x - H_{j-1,k}^x\right) - n_y^1\left(H_{j-1,k-1}^y - H_{j-1,k}^y\right)\right) \\
&+\frac{\cos\bar{\phi}_j}{\epsilon c}\frac{\partial}{\partial t}\left(n_x^1\left(H_{j,k}^x - H_{j,k-1}^x\right) + n_y^1\left(H_{j,k}^y - H_{j,k-1}^y\right)\right) \\
&+\frac{\sin^2\bar{\phi}_j}{\epsilon c T\cos\bar{\phi}_j}\left(n_x^1\left(H_{j,k}^x - H_{j,k-1}^x\right) + n_y^1\left(H_{j,k}^y - H_{j,k-1}^y\right)\right) \\
&+\frac{\cos\phi_k}{\epsilon c}\frac{\partial}{\partial t}\left(n_x^2\left(H_{j-1,k-1}^x - H_{j,k-1}^x\right) + n_y^2\left(H_{j-1,k-1}^y - H_{j,k-1}^y\right)\right) \\
&+\frac{\sin^2\phi_k}{\epsilon c T\cos\phi_k}\left(n_x^2\left(H_{j-1,k-1}^x - H_{j,k-1}^x\right) + n_y^2\left(H_{j-1,k-1}^y - H_{j,k-1}^y\right)\right) \\
&+\frac{\cos\bar{\phi}_k}{\epsilon c}\frac{\partial}{\partial t}\left(-n_x^2\left(H_{j,k}^x - H_{j-1,k}^x\right) - n_y^2\left(H_{j,k}^y - H_{j-1,k}^y\right)\right) \\
&+\frac{\sin^2\bar{\phi}_k}{\epsilon c T\cos\bar{\phi}_k}\left(-n_x^2\left(H_{j,k}^x - H_{j-1,k}^x\right) - n_y^2\left(H_{j,k}^y - H_{j-1,k}^y\right)\right) = 0
\end{aligned} \tag{27}$$

We have now written down $2P(P+1) + P^2 = 3P^2 + 2P$ equations in $3(P+1)^2 = 3P^2 + 6P + 3$ variables. Thus $4P+3$ additional equations are required. We first incorporate incoming data from the edges. Here we define outgoing characteristics (ingoing to the corner) in the tangential directions for each edge. Then for $k = 0, \ldots, P$ set

$$S_k^1 = E_{0,k}^z + s_1\sqrt{\frac{\mu}{\epsilon}}\left(n_x^1 H_{0,k}^x + n_y^1 H_{0,k}^y\right). \tag{28}$$

Then with the correspondence

$$\left(E_{0,k}^z, H_{0,k}^x, H_{0,k}^y\right) \leftrightarrow \left(E_k^z, H_k^x, H_k^y\right)^{\text{edge}_1} \tag{29}$$

we can impose

$$\frac{\partial S_k^1}{\partial t} = \left(\frac{\partial S_k^1}{\partial t}\right)^{\text{edge}_1}. \tag{30}$$

Similarly for $j = 0, \ldots, P$ set

$$S_j^2 = E_{j,0}^z + s_2 \sqrt{\frac{\mu}{\epsilon}} \left(n_x^2 H_{j,0}^x + n_y^2 H_{j,0}^y \right), \tag{31}$$

and note the correspondence

$$\left(E_{j,0}^z, H_{j,0}^x, H_{j,0}^y \right) \leftrightarrow \left(E_j^z, H_j^x, H_j^y \right)^{\text{edge}_2}. \tag{32}$$

Then we can impose

$$\frac{\partial S_j^2}{\partial t} = \left(\frac{\partial S_j^2}{\partial t} \right)^{\text{edge}_2}. \tag{33}$$

Finally we can impose the termination conditions

$$R_{+,j,P}^1 = 0 \quad j = 0, \ldots, P \tag{34}$$

$$R_{+,P,k}^2 = 0 \quad k = 0, \ldots, P.. \tag{35}$$

These constitute $4(P+1) = 4P+4$ equations, one more than we can use. We remove an equation by only imposing the sum of (30) and (33) for $j = k = 0$.

4 Meep

Meep-1.2 and related supporting software, ***libctl-3.2.1***, ***harminv***, ***harminv-devel*** have been installed in Linux systems. Single CPU build has been tested and verified against the examples posted on the Meep website. Parallel version has not been tested yet.

In Fedora OS systems, default configure for libctl-3.2.1 will not work, instead, we need to use
./configure LDFLAGS=-lm
to configure the software.

Packages ***harminv*** and ***harminv-devel*** will be needed for solutions' Fourier transforms and need to be installed before compile Meep.

Default configure for Meep-1.2 also failed in our Linux system. It cannot automatically include the Lapack library when trying to link the software. We need to explicitly add the Lapack library as follows:
./configure LDFLAGS=-llapack prefix=$Meep_home
with ***$Meep_home*** the destination of Meep installation.

Post-processing utility ***h5utils-1.12.1*** is also installed. There are a lot of utilities included in this package. To manipulate unsteady solution easier, we developed a python script to call **h5tovtk** utility and create a series of .vtk file from the big .h5 file. The script is list below

```
#!/usr/bin/python
import commands
import string
import sys

argc = len(sys.argv)

if argc != 2:
  print "usage error, wrong number of arguments"
  print "usage example: ./convH5ToVtk.py sq_scatter-hz.h5"
```

```
  exit(-1)

#get the final time step
cmd = "h5ls " + sys.argv[argc-1]
h5lso = commands.getoutput(cmd)
h5lso = h5lso.replace("/Inf}", "")
lspace = h5lso.rfind(" ")
tend = int(h5lso[lspace+1:])
print "final time step is: ", tend-1

fileRoot = sys.argv[argc-1]
fileRoot = fileRoot[:len(fileRoot)-3]
defaultVtk = fileRoot+".vtk"
print fileRoot

for i in range(tend):
  bigNum = 1000000+i
  newVtk = fileRoot+"_"+(str(bigNum))[1:]+".vtk"
  cmd1 = "h5tovtk -t " + str(i) + " " + sys.argv[argc-1]
  commands.getoutput(cmd1)
  cmd2 = "mv " + defaultVtk + " " + newVtk
  commands.getoutput(cmd2)
  print newVtk + " is ready"
```

With .vtk files, we can easily use **paraview** to do more detailed analysis.

Meep code was verified with several examples using the input data provided in the **Meep Tutorial** web page. Grid resolution and pml-layer thickness have been studied to check the code accuracy and pml-layer sensitivity.

Figure 1 gives the solution of E_z 2-D field in a a straight waveguide. The waveguide has $\epsilon = 12$. Figure 2 depicts the solution of a 90^o bend waveguide solution at $t = 79.8, 139.8$ and 199.8 respectively. To further check the code capability, a 3D case with a perfect-electric-conductor sphere put at the center of the domain is calculated. The wavelength is set to $\lambda = 1m$, and the source frequency nondimensional frequency $f = 1$ (which indicates the real frequency $= f * c/\lambda = 3 \times 10^8$ Hz). The domain is set to $10 \times 10 \times 10$, and the radius of the sphere is set to $r = 10\lambda/(2\pi) = 1.59m$. Figure 3 shows the E_z field distribution at non-dimensional $t = 200$. For this run, with resolution=20, it takes about 2 hours in an Intel core-i7 CPU.

In Figure 2 , we computed the field patterns for light propagating around a waveguide bend. The results are only visually pretty and not quantitatively satisfying. We'd like to know exactly how much power makes it around the bend, how much is reflected, and how much is radiated away. This is done by running the example case *bend-flux* in the Meep example website. The domain size is 16×32, including the pml-layers, which we set the thickness to be 1. We follow the instruction in the tutorial, and run with resolution 10, 20 and 40. Figure 4 depicts the comparison of the case with different resolution. It can be seen a minimum resolution of 20 is needed.

In addition, we tried different pml thickness with resolution 20. Figure 5 depicts the comparison of the case with different pml thickness. This is to check the robustness of the radiation BC in Meep code and see how much we can improve by using CRBC module in this project in the future. Obviously, with very thin pml, the results oscillate, which is in our expectation. However, when pml thickness increased to 2, the solution is also far away from the thickness 1 benchmark, which we may need to find out later.

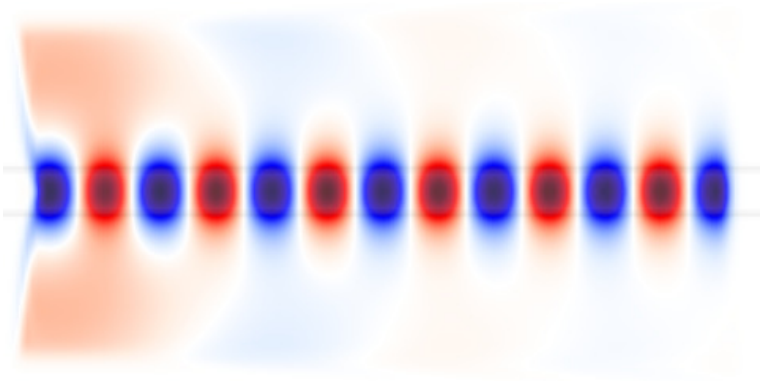
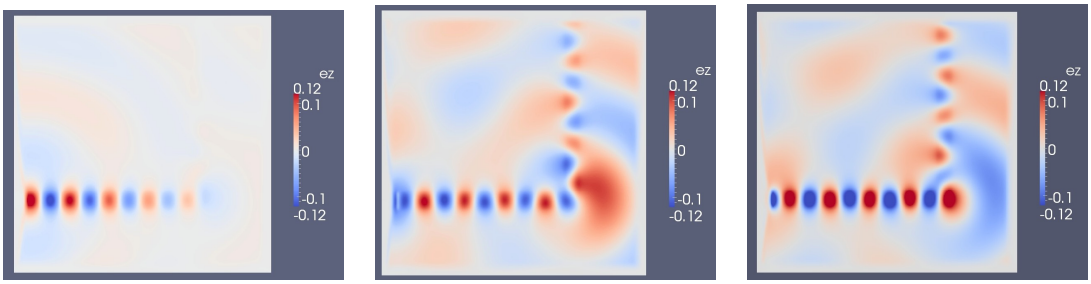


Figure 1: Electric-field component E_z distribution of straight waveguide field at $t = 200$



(a) (b) (c)

Figure 2: Electric-field component E_z distribution at of the 90^o waveguide field at $t = 79.8, 139.8$ and 199.8.

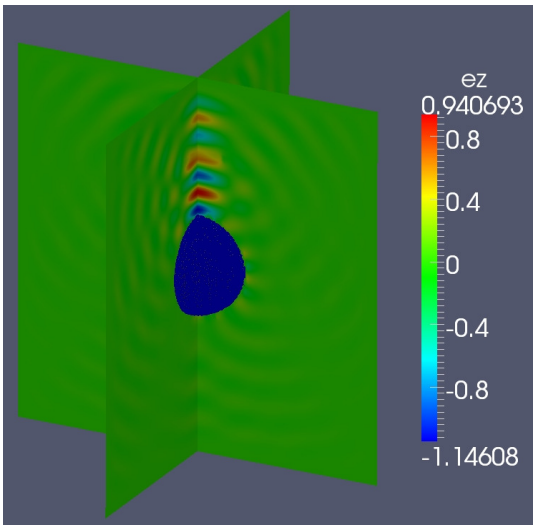


Figure 3: Electric-field component E_z distribution of a 3D case at $t = 200$

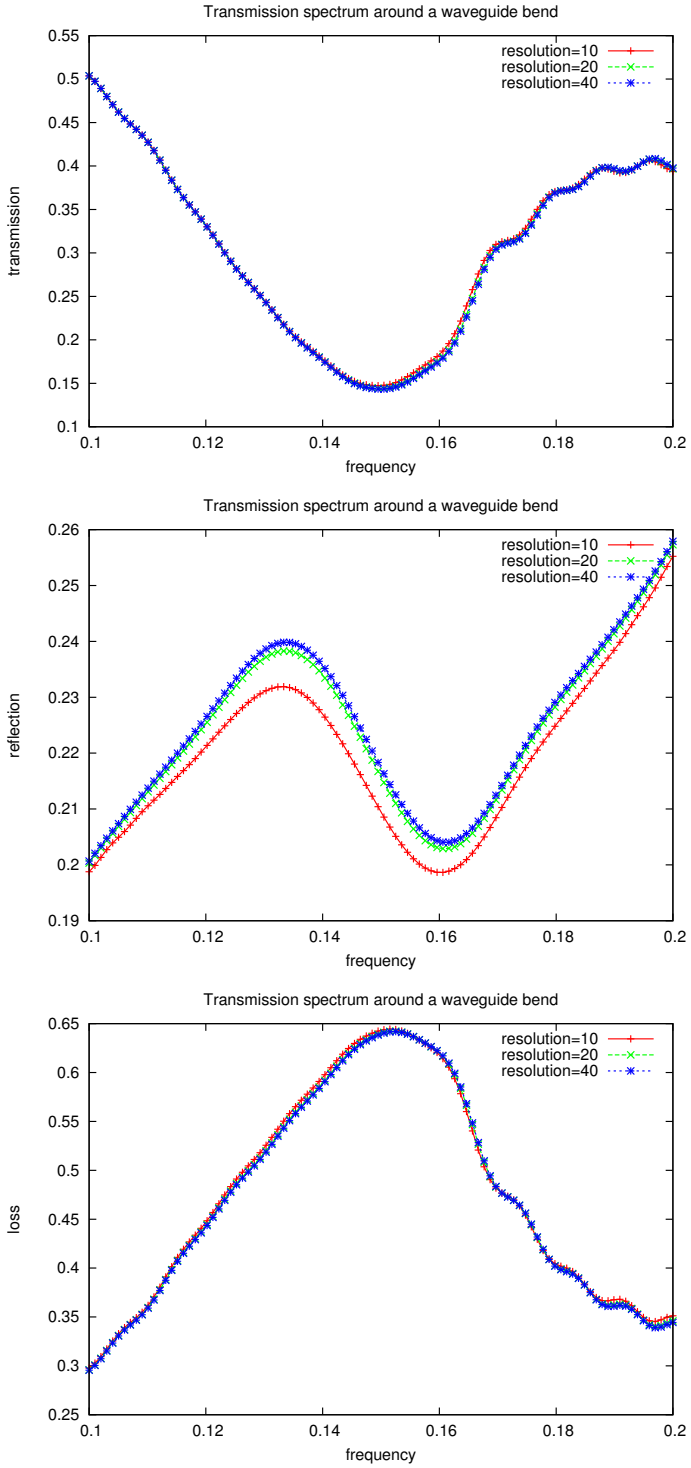


Figure 4: Transmission spectrum around a waveguide bend using different grid resolution.

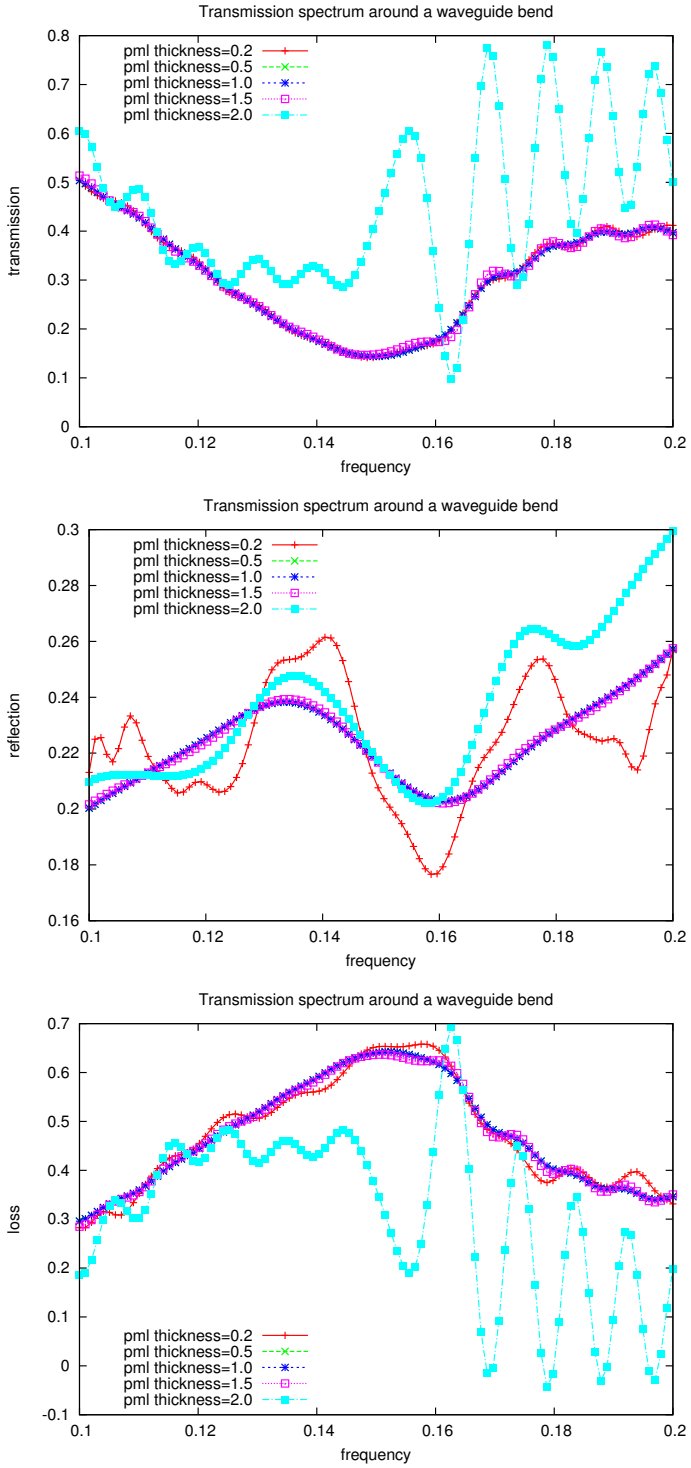


Figure 5: Transmission spectrum around a waveguide bend using different pml thickness.

References